Priberam's question answering system for Portuguese

Carlos Amaral, Helena Figueira,
André Martins, Afonso Mendes, Pedro Mendes, Cláudia Pinto

Priberam Informática
Av. Defensores de Chaves, 32 - 3º Esq.
1000-119 Lisboa, Portugal
Tel.: +351 21 781 72 60
Fax: +351 21 781 72 79

{cma, hgf, atm, amm, prm, cp}@priberam.pt

Abstract

This paper describes the work done by Priberam in the development of a question answering (QA) system for Portuguese. The system was built using the company's NLP workbench and information retrieval technology. Special focus is given to the question analysis, document and sentence retrieval, and answer extraction stages. The paper discusses the system's performance in the context of the QA@CLEF 2005 evaluation.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2 [**Database Management**]: H.2.3 Languages—*Query Languages*

General Terms

Measurement, Performance, Experimentation, Languages

Keywords

Question answering, Questions beyond factoids

1 Introduction

The 2004 CLEF campaign introduced Portuguese as one of the working languages [1] and allowed the evaluation of two monolingual question answering (QA) systems for European Portuguese [2, 3]. In 2005, the organization added new resources, making both European Portuguese and Brazilian Portuguese available to CLEF participants. The set of target documents now comprises the collection of news published during the years 1994 and 1995 by the Portuguese newspaper *Público* and by the Brazilian newspaper *Folha de São Paulo*. The test set includes 200 questions also in European and Brazilian Portuguese.

Our approach to this year's QA track at CLEF (QA@CLEF) relies on previous work done for the Portuguese module of TRUST – Text Retrieval Using Semantic Technologies[1] –, an European Commission co-financed project[2] whose aim was the development of a multilingual semantic search

[1] See `http://www.trustsemantics.com`.
[2] Cooperative Research (CRAFT) project number IST-1999-56416.

engine capable of processing and answering natural language questions in English, French, Italian, Polish and Portuguese [4, 5]. In the TRUST project, the system searches a set of plain text documents (either in a local hard disk or in the Web) and returns a ranked list of sentences containing the answer to a given natural language question. The goal of QA@CLEF is similar, except that it must extract a unique exact answer from the retrieved sentences.

The architecture of our QA system is built upon a standard approach. After the question is submitted, it is categorized according to our question typology and, through an internal query, a set of potentially relevant documents is retrieved. Each document contains a list of sentences which were assigned the same category as the question. Sentences are weighted according to their semantic relevance and similarity with the question. Next, through specific answer patterns, these sentences are again examined and the parts containing possible answers are extracted and weighted. Finally, a single answer is chosen among all candidates.

In the next section, we address the various tools and resources developed or used in the system's underlying natural language processing (NLP). Section 3 provides an overview of the QA engine architecture, namely the indexing process, the question analysis, the document and sentence retrieval procedures and the answer extraction. Section 4 details the experimental results of our system in QA@CLEF, and section 5 presents our conclusions and guidelines for future work.

2 A workbench for NLP

Previous work on the development of linguistic technology for FLiP, *Ferramentas para a Língua Portuguesa*, Priberam's proofing tools package for Portuguese[3], as well as on the construction of the Portuguese module of the already mentioned TRUST search engine, required the development of a workbench for NLP [6]. This workbench includes lexical resources, software tools, statistical information extracted from corpora, contextual rules, and other tools and resources adapted to the task of question answering.

2.1 Lexical resources

Our lexical resources include several lexical databases, such as a wide coverage lexicon, a thesaurus and a multilingual ontology.

The lexicon comprises, for each lexical unit, information about part of speech (POS), sense definitions, semantic features, subcategorization and selection restrictions, ontological and terminological domains, English and French equivalents and lexical-semantic relations. For our QA@CLEF monolingual task, we do not use the English and French equivalents, whose purpose is essentially to perform cross-language tasks.

The thesaurus provides a set of synonyms for each lexical unit, allowing, by means of query expansion, to improve the information retrieval stage by including documents and sentences that contain synonyms of the question's keywords.

Another major lexical component of the workbench is the multilingual ontology, which groups words and expressions through their conceptual domains. It was initially designed by Synapse Développement, the French partner of TRUST, and then converted into all the languages of the consortium[4]. The combination of the ontologies of all TRUST languages provides a bidirectional word/expression translation mechanism, having the English language as an intermediate. It is thus possible to operate in a cross-language environment, allowing, for instance, to obtain answers in French for questions formulated in Portuguese, or vice-versa. Synapse Développement carried out such an experiment and submitted a Portuguese-French run to this year's bilingual task of QA@CLEF [7], making use of Priberam's TRUST Portuguese module to analyse the test set of questions.

[3] FLiP includes a grammar checker, a spell checker, a thesaurus and a hyphenator that enable different proofing levels – word, sentence, paragraph and text – of European and Brazilian Portuguese. An online version is available at `http://www.flip.pt`.

[4] The ontology is designed to incorporate additional languages in future projects.

Additionally, lexical resources include question identifiers, *i.e.*, semantically labelled words for question categorization. These are groups of words related with typical questions domains and sub-grouped according to their POS. For instance, the label <DIMENSION> includes measuring units (with their abbreviations or symbols), nouns, adjectives and verbs related with dimension, distance and measurement.

2.2 Software tools

The lexical resources just described interact with software tools that we have implemented, like *SintaGest* program. Priberam's SintaGest is an interactive tool that allows building and testing a grammar for any language; it was successfully used by the company's linguistic and programming teams to develop European and Brazilian Portuguese grammars. SintaGest allows a practical way to code transformation rules for morphological disambiguation and named entity recognition, as well as production rules to build a context-free grammar (CFG). In addition, it allows to perform tasks related with QA, such as writing patterns to categorize questions and extract answers. After being tested, these rules are compiled to generate compressed and optimized low-level information. Furthermore, SintaGest can also run in batch mode on a corpus, to test the grammar, generate reports, extract collocations and named entities, collect statistical information, etc. For a detailed description of some of these SintaGest features, see again [6].

Along with SintaGest, several modules have been developed to perform more specific tasks. One of such tasks is morphological disambiguation. It is done in two stages: first, the contextual rules defined in SintaGest are applied; then, remaining ambiguities are suppressed with a statistical POS tagger based on a second-order hidden Markov model (HMM). This turns out to be a fast and efficient approach using the Viterbi algorithm [8, 9]. The prior contextual and lexical probabilities were estimated by processing large, partially tagged corpora, among them the *CETEMPúblico 1.7* collection of news from the Portuguese newspaper *Público*[5]. Lexical probabilities are encoded for each lemma, rather than for each word. To achieve this, we calculated, for each lemma, its frequency and the relative frequency of its inflections. Then, those lemmas with similar distributions for their inflections are grouped into a smaller number of classes. Clustering techniques based on competitive learning [10] are used to choose the number of classes, group the lemmas and characterize each class. Working with these clusters is advantageous because, on one hand, we can extend the behaviour to words that are not so frequent in our corpora, and, on the other hand, we can compress the information we need at runtime.

2.3 Contextual rules

As said above, SintaGest provides a way to build contextual rules for performing morphological disambiguation, named entity (NE) recognition, etc. An editor allows writing, compiling and testing these rules. Once validated, they are then used in our QA system at runtime.

NEs appear frequently both in questions and in the texts to index. They can be proper nouns of organizations, places, event dates, etc. Besides NEs, some expressions (e.g. nominal, adjectival, verbal and adverbial phrases, including the dates in temporally restricted questions) are frequent and idiomatic enough to justify their handling as if they were single tokens.

The NE recognizer is capable of detecting and tagging a large amount of NEs. The tagger tries to find a sequence of proper nouns, recognizing it as a single token and classifies syntactically and semantically the NE thus created, namely by inheriting the features of its head (e.g. *Luís Vaz de Camões* will be classified as an anthroponym, since *Luís* is classified in the lexicon as such). It also uses groups of conceptually gathered words that will help in the classification of NE: for instance, a sequence of proper nouns preceded by a common noun such as *rio* [river] will be classified as a toponym (e.g. *rio de São Domingos* [São Domingos river]). For semantic disambiguation purposes, the NE recognizer also considers the context, checking what words precede or follow the NE.

[5] Available at `http://acdc.linguateca.pt/cetempublico`.

2.4 Question categorization

Classifying questions into categories is a key task during question analysis, since it allows filtering out unrelated documents and applying more tuned extraction rules in the candidate sentences. To address this, we used a set of 86 question categories previously defined for TRUST by Synapse Développement. Table 1 illustrates some of the categories currently used in our QA system.

Category	Example
<DENOMINATION>	"Nomeie um cetáceo." [Name a cetacean.]
<DATE OF EVENT>	"Em que dia foi inaugurada a Torre Eiffel?" [On what day was the Eiffel Tower inaugurated?]
<TOWN NAME>	"Em que cidade fica o campo de concentração de Auschwitz?" [In what city is the Auschwitz concentration camp located?]
<BIRTH DATE>	"Quando nasceu a ovelha Dolly?" [When was sheep Dolly born?]
<FUNCTION>	"Quem é Jorge Sampaio?" [Who is Jorge Sampaio?]

Table 1: Example of categories of question.

Once the categories are defined, a way must be provided to *categorize*, *i.e.*, to automatically select one or more categories to a given question. Common approaches involve writing simple patterns, using for instance regular expressions [11], and optionally complement them with rules obtained through some sort of supervised learning in a large training set [12]. We discarded any learning-based method since the training set should be large enough to offer an adequate coverage of all the categories, which in our case are numerous. As for methods based on regular expressions, they have the disadvantage of being too focused on string patterns, discarding other useful features, and thus leading to a relatively small coverage of instances of question. Our approach tries to overcome the above limitations by using patterns that are much more powerful than regular expressions. Like with contextual rules, SintaGest provides the interface for writing, testing and compiling such patterns. They were tested and validated with real world questions through the *CLEF Multieight-04 Corpus of 700 questions and manually retrieved answers*[6] [13]. Each pattern is a sequence of 'terms' with the possible types listed in Table 2 (prefix `Any` is used to build disjunctive terms).

Terms may be conjugated (e.g. `Word(casa) & Cat(N)` means that the current word should be the common noun *casa* [house], and not a form of the verb *casar* [to marry]). Besides, a term may also be optional (e.g. `Word(casa)?` means that the presence of the word *casa* in the current position is optional), and distances between terms may be defined (e.g. `Word(quem) Distance(1,3) Word(presidente)` means that between the words *quem* [who] and *presidente* [president] there can be a minimum of 1 and a maximum of 3 words.

Patterns built with these terms are used not only to categorize questions, but also to categorize general sentences and even to extract answers. Actually, there are 3 kinds of patterns:

- *Question patterns* (QPs) are used to assign categories to questions. More than one category per question is allowed, thus avoiding difficulties in choosing the most suitable category.
- *Answer patterns* (APs) are used to assign categories to a general sentence during the indexation stage, which means that the sentence contains possible answers for questions with those categories. Again, more than one category per sentence is allowed.
- *Question answering patterns* (QAPs) are used to extract a possible answer for a specific question.

[6] Available at `http://clef-qa.itc.it/2005`.

Type of Term	Meaning	Example	Description
Word	Literal word or expression	Word(presidente)	Current word or expression is *presidente* [president]
		AnyWord(presidente, chefe de estado)	Current word or expression is either *presidente* [president] or *chefe de estado* [head of state]
Root	Lemma	Root(presidente)	Current lemma is *presidente* [president]
		AnyRoot(presidente, chefe de estado, primeiro-ministro)	Current lemma is either *presidente* [president], *chefe de estado* [head of state] or *primeiro-ministro* [prime minister]
Cat	POS tag with optional features	Cat(N(MASC,SING,,,,ERG))	Current POS is a common noun, masculine, singular and an ergonym[7]
		AnyCat(N, Nprop, ENT)	Current POS is either a common or proper noun or a named entity
Ont	Ontology entry	Ont(100.3)	Current word or expression is part of the ontology level 100.3 (Colours/grey)
		AnyOnt(100.1, 100.2, 100.3)	Current word or expression is part of the ontology levels 100.1 (Colours/white), 100.2 (Colours/black) or 100.3 (Colours/grey)
QuestIdent	Question identifier	QuestIdent(FUNCTION_N)	Current word is a noun/noun phrase question identifier for the <FUNCTION> category
		AnyQuestIdent(FUNCTION_N, FUNCTION_ADJ)	Current word is either a noun/noun phrase or adjective/adjective phrase question identifier for the <FUNCTION> category
Const	Constant previously defined	Ergonym	Current word is the beginning of a phrase with an ergonym
		AnyConst(Ergonym, NounPhrase)	Current word is the beginning of a phrase with an ergonym or a noun phrase

Table 2: Types of terms used in patterns.

When a question is posed and it matches a QP, a category is assigned for the question and a set of QAPs becomes active. Then, documents containing sentences with categories in common with the question (previously determined during indexation via the APs) are analysed; the active QAPs are then applied to each sentence in order to extract the possible answers.

Table 3 shows examples of QPs, APs and QAPs for the category <FUNCTION>. There are two blocks of code: the question answer block is checked when the QA system is answering questions, while the answer block is only used for indexation of the document collection. Each pattern has a score following the = sign to establish a priority and to reflect how likely it can match a true sentence. Of course, these scores must be heuristically adjusted in order to give preference to more reliable and more specific patterns. Optional terms that are verified are rewarded adding 3 units to the pattern score. Distances penalize the pattern score by subtracting as many units as the

[7]The word *ergonym* (from Greek *ergon* 'work' and *onoma* 'name') designates here a person's profession, job, function, post, etc..

difference to the specified minimum distance. The `With` command between terms means that the second term must be verified somewhere inside the first term, usually a constant that defines a phrase. Finally, notice that QAPs include an extra term, named `Pivot`, to signal keywords that are present both in the question and in the matched sentence (see subsection 3.2 for details), as well as a sequence of terms delimited by curly brackets, to signal the words that are to be extracted as a possible answer.

```
// Example of a question answer block encoding QPs and QAPs:

Question (FUNCTION)
     :  Word(quem) Distance(0,3) Root(ser) AnyCat(Nprop, ENT) = 15
        // e.g.  ``Quem é Jorge Sampaio?''
     :  Word(que) QuestIdent(FUNCTION_N) Distance(0,3) QuestIdent(FUNCTION_V) = 15
        // e.g.  ``Que cargo desempenha Jorge Sampaio?''
Answer
     :  Pivot & AnyCat (Nprop, ENT) Root(ser) Definition With Ergonym?  = 20
        // e.g.  ``Jorge Sampaio é o {Presidente da República}...''
     :  {NounPhrase With Ergonym?} AnyCat (Trav, Vg) Pivot & AnyCat (Nprop, ENT) = 15
        // e.g.  ``O {presidente da República}, Jorge Sampaio...''
     ;
```

```
// Example of an answer block encoding APs:

Answer (FUNCTION)
     :  QuestIdent(FUNCTION_N) = 10
     :  Ergonym = 10
     ;
```

Table 3: Examples of patterns.

Current work is being made to add new features to these patterns. One of the features being developed is a new type of term for syntactic phrases, more powerful than the current `Const` term. This feature is essential for the improvement of question categories like <AIM>, <CAUSE>, <CONSEQUENCE> or <CONDITION>, which require general syntactical patterns for extraction of possible answers specifically in adverbial subordinate clauses. Other features involve enhancing the QPs syntax to encode a measure of importance for the question pivots and to embed sense disambiguation rules. This would allow to perform word sense disambiguation during question analysis and thus to select a stricter set of relevant ontology levels and synonyms that will be searched during the document retrieval stage.

3 System description

The architecture of our QA system is fairly standard. It involves five major tasks, described in the current section: (*i*) the indexing process, (*ii*) the question analysis, (*iii*) the document retrieval, (*iv*) the sentence retrieval, and (*v*) the answer extraction.

3.1 Indexing Process

The indexation is an off-line procedure by which a set of target documents is parsed in order to collect information in index files. Previous work on this subject has been done during the development of LegiX, Priberam's juridical information system[8]. The indexing engine of LegiX

[8]For more information about LegiX, see `http://www.legix.pt`.

was adapted to index semantic information, ontology domains, question categories and other specificities for QA.

In the case of the Portuguese target collection of QA@CLEF there was a total of 210734 indexed documents. For each document, we collected its most relevant ontological and terminological domains and, for each sentence, the question categories for which it contains possible answers, determined through the APs referred in subsection 2.4. After applying morphological disambiguation (see the last paragraph of subsection 2.2 for a description of how it is made), we collect as key elements for indexation, the words of each sentence that are not considered stop words. Each word is represented by a unique triple {lemma, head of derivation, POS}. Special words as numbers, dates, fixed expressions, NEs and proper nouns are flagged. Multiple word expressions (e.g. NEs) are indexed as well as each word that composes them. Unlike the system used in the TRUST project, here we chose not to perform word sense disambiguation (WSD). We justify this decision with the following reasons: (*i*) our current WSD is still at an early stage and has a poor performance, and (*ii*) making automatic WSD during question analysis is inherently a difficult task. Indeed, in TRUST the user performs manually the disambiguation at this stage by selecting the appropriate sense of each word of the question. As stated in the last paragraph of subsection 2.4, we intend to develop a scheme to embed WSD in the QPs, since these patterns usually reduce the context scope, making the task less difficult to achieve.

For performance reasons, each word in the index is stored with a reference not only to the target documents in which it occurs, but also to the sentences indices inside each document. This accelerates the document retrieval stage, as we describe in subsection 3.3.

3.2 Question analysis

Since indexation is performed off-line, the question analyser is indeed the first module of our system. It receives as input a NL question q submitted by the user, that is first lemmatized and morphologically disambiguated (see subsection 2.2). The next step consists on interpreting it.

Like the majority of the approaches, we start by categorization. In fact, results show clearly that determining the domain of the question and characterizing the desired format for the answer is an essential step in QA systems. However, approaches diverge about the number, structure (flat or hierarchized), and choice of the categories (see [14, 15] for interesting discussions on this matter). As described in subsection 2.4, we use 86 categories in a flat structure and build powerful QPs to categorize the questions, instead of the commonly used patterns based on regular expressions. When this categorization stage ends, the following information has been gathered: (*i*) one or more question categories, $\{c^1, c^2, \ldots, c^m\}$, (*ii*) a list of active QAPs (see subsection 2.4) to be later applied during answer extraction (see subsection 3.5), and (*iii*) a score σ^{QP} for each question pattern that matched the question.

We next proceed to the extraction of pivots. Pivots are the key elements of the question, and they can be words, expressions, NEs, phrases, numbers, dates, abbreviations, etc.. For each pivot, we collect the word or words that make the pivot itself, its lemma w_L, its head of derivation w_H, its POS, their synonyms $w_S^1, \ldots, w_S^n$ provided by the thesaurus (subsection 2.1), and flags to indicate if they are special words. Together with the above mentioned question categories, the relevant ontological and terminological domains in the question, $\{o^1, o^2, \ldots, o^p\}$, are also collected.

This data then feeds the document retrieval module, described in the next subsection.

3.3 Document retrieval

After analysing the question, we submit a query to the index files using as search keys the pivot lemmas, their heads of derivation, their synonyms, the ontological domains and the question categories.

Let w_L^i, w_H^i and $w_S^{i,j}$ denote respectively the i-th pivot lemma, its head of derivation and its j-th synonym. Each of these synonyms has a weight $\rho(w_S^{i,j}, w_L^i)$ to reflect its semantic proximity with the original pivot lemma w_L^i. In the following, we denote by c^i and o^i the i-th possible category for the posed question and the j-th relevant ontological or terminological domain, respectively.

For each word, we calculate a weight $\alpha(w)$ given by:

$$\alpha(w) = \alpha_{POS}(w) + K_{ilf} ilf(w) + K_{idf} idf(w) \tag{1}$$

In (1), the α_{POS} is used to reflect the influence of the POS on the pivot's relevance. For instance, since we consider that pivots that are NEs, are generally more important than common nouns, and these than adjectives or verbs, we have a chain $\alpha_{POS}(NE) \geq \alpha_{POS}(N) \geq \alpha_{POS}(ADJ) \geq \alpha_{POS}(V)$. Of course, these are general assumptions: there are many questions where a verb is more relevant than an adjective, although the opposite situation is slightly more frequent (for example, in a question like "Como se chama o primeiro presidente americano?" [What is the name of the first American president?] the adjectives *primeiro* and *americano* are much more important than the verb *chamar*). As briefly stated in the last paragraph of section 2.4, we intend to introduce here a new parameter to express the importance of each pivot, eventually taking into account the syntactic parsing of the question. Yet in (1), K_{ilf} and K_{idf} are fixed parameters for interpolation, while ilf and idf denote respectively the *inverse lexical frequency* – that is, the logarithm of the inverted relative frequency of the word in the corpus – and the commonly used inverse document frequency (see [16] for an explanation). We opted not to include a tf term for the word frequency in the document, because of the relatively small size of each document.

Consider now the document collection. Let d be a particular document and define $\delta_L(d, w_L) = 1$ if d contains the lemma w_L and 0 otherwise. Moreover, define $\delta_H(d, w_H)$ in the same way for the head of derivation w_H, and $\delta_C(d, c)$ and $\delta_O(d, o)$ analogously for the question category c and the ontological domain o. We calculate the document score σ^d as:

$$\begin{aligned} \sigma^d \quad = \quad & \sum_i \max \left\{ K_L \delta_L(d, w_L^i)\alpha(w_L^i), K_H \delta_H(d, w_H^i)\alpha(w_H^i), \max_j K_S \delta_L(d, w_S^{i,j})\alpha(w_S^{i,j})\rho(w_S^{i,j}, w_L^i) \right\} \\ & + K_C \max_i \delta_C(d, c^i) + K_O \max_i \delta_O(d, o^i), \end{aligned} \tag{2}$$

where K_L, K_H, K_S, K_C and K_O are fixed scaling constants with $K_L > K_H > K_S$ to reward matches of lemmas, that are stronger than those of heads of derivation and synonyms.

The score in (2) is then fine-tuned to take into account the pivot proximity in the documents, rewarding those in which the pivots occur in sentences close together. At the end, the top 30 documents are retrieved to be further analysed at sentence level. In order to avoid the need of analysing the whole text, each document contains a list of indices of sentences where the above pivot matches occurred.

3.4 Sentence retrieval

This module receives as input a set of documents, whose sentences that match the pivots are marked. Our engine allows to analyse not only these sentences, but also the k sentences before and after, where k is configurable. However, making use of this feature could cause processing in this stage to become too heavy, especially in situations where many documents with many marked sentences are retrieved. Besides, to take full profit of this, additional techniques would be required to find connections among close sentences, for instance through anaphora resolution. Hence, for now we simply set $k = 0$.

Let s be a particular sentence to be analysed at this stage. After parsing s, we calculate a score σ^s taking into account:

- The number of pivots matching s;
- The number of pivots having in common the lemma or the head of derivation with some token in s;
- The number of pivot synonyms matching s;
- The order and proximity of the pivots in s;

- The existence of common question categories between q and s;
- The number of ontological and terminological domains characterizing q which are also present in s;
- The score σ^d of the document d that contains s.

Here, partial matches are also considered: for instance, if only a word of a given NE is found in the sentence (e.g. *Fidel* of the anthroponym *Fidel Castro*), then it will contribute with a lower weight than if it was a complete match.

To save efforts in the subsequent answer extraction module, sentences s that are scored below a fixed threshold or where the total number of matches (either complete or partial) is lower than a fixed fraction of the total number of pivots are immediately discarded. The remaining sentences and their scores are passed as output to the next module.

3.5 Answer extraction

The answer extractor receives as input a set $\{s, \sigma^s\}$ of scored sentences presumably containing answers. Each of these sentences is then tested against the QAPs that were activated during the question analysis stage (see subsection 3.2). Notice that these QAPs are directly linked with the QP that matched the question (see Table 3). As said in subsection 2.4, each QAP includes information on what part of the sentence (if any) is to be extracted as a possible answer; it also has a score to reflect the relevance of the QAP and the pertinence of the foreseen answer.

Let us suppose that a particular sentence s matches a specific QAP. The curly bracketed terms in the QAP extract one or more candidate answers from s (notice that a single pattern can match s in several different ways). When all the active QAPs are applied, we are led to zero or more possible answers extracted from s. Answers that are substrings of others are discarded, unless they have a higher score: this tends to privilege longer answers. In specific cases, the opposite behaviour can be forced by properly setting the scores. Answers containing question pivots are not allowed, unless they are part of NEs (e.g. *Deng Nan* is allowed as an answer to the question "Como se chama a filha de Deng Xiao Ping?" [What is the name of Deng Xiao Ping's daughter?], while *filha* is not). Suppose that a sentence s matches some QAP with score σ^{QAP}, linked with a QP with score σ^{QP}, such that a is extracted from s and becomes a candidate answer. In this scenario, a will have the following score σ^a assigned:

$$\sigma^a = K_s\sigma^s + K_{QP}\sigma^{QP} + K_{QAP}\sigma^{QAP} + \sum \sigma^{rew} - \sum \sigma^{pen} \tag{3}$$

In (3), K_s, K_{QP} and K_{QAP} are interpolating constants and $\sum \sigma^{rew} - \sum \sigma^{pen}$ is the total amount of rewards minus the total amount of penalties applied when processing the QAP. These rewards and penalties are small quantities usually due to optional terms and variable distances in the QAP (see subsection 2.4 for a further explanation).

The last step consists in analysing all the answer candidates $\{a, \sigma^a\}$, if any, and choosing the best one as the final answer. If none has been chosen, "NIL" will be displayed. To accomplish this, the answer scores $\{\sigma^a\}$ are first adjusted with additional rewards to take into account the repeatability of the words of each answer in the collection of answer candidates that were extracted from sentences scored above a fixed threshold; this threshold avoids the repeatability of erroneous answers.

In the end, the system outputs the answer with the highest score, $\hat{a} = \arg\max_a \sigma^a$, or "NIL" if none is available. Currently, no confidence score is being measured and no further verification is made to check if $\hat{a}$ really answers the question posed q. This is something to be done in the future.

4 Results

The test set of 200 questions run in the monolingual task covered mainly factoid questions (158 in all) and a few (42) definition questions. Table 4 presents the scores of the submitted run.

Question ↓ **Answer →**	Right	Wrong	Inexact	Unsup.	**Total**	**Accuracy (%)**
Factoid (F)	91	38	5	1	135	67.4
Definition (D)	27	7	8	0	42	64.2
Temporally restricted factoid (T)	11	10	0	2	23	47.8
Total	129	55	13	3	200	**64.5**

Table 4: Results by type of question.

The F-questions and D-questions statistics add to a satisfactory overall accuracy of the system, whose performance is comparable to that of the best scored systems in recent evaluation campaigns [17].

Several reasons contribute to the lower accuracy of T-questions. Firstly, we do not index dates differently from other keywords. For instance, *25 de Abril de 1974* and *25/4/1974* are currently indexed as different terms, and as a result, they cannot match during sentence retrieval and answer extraction stages. Because of this limitation, we do not force the date to be present in the sentence from which we extract the answer. This, in turn, leads to inexact answers that would be correct if the question was not temporally restricted. We also do not take into account the documents dates when answering T-questions. For instance, in question 3 "Quantos capacetes azuis holandeses havia em Srebrenica, na Bósnia, em Julho de 1995?" [How many Dutch blue helmets were there in Srebrenica, Bosnia, in July 1995?] we returned, during document retrieval, a few documents dated before July 1995, which could not contain a valid and supported answer. These aspects will be taken into account in a near future: we plan to index dates (and numbers) in a proper numeric format, using documents dates to filter out the obsolete ones, as well as converting relative temporal references (like *ontem* [yesterday]) into absolute ones.

Table 5 summarizes the main tasks where the system failed.

Tasks ↓ **Question →**	F-quest.	D-quest.	T-quest.	**Total**	**Failure (%)**
Document retrieval	6	2	1	9	4.5
Extraction of candidate answers	18	7	8	33	16.5
Choice of the final answer	7	5	1	13	6.5
NIL validation	12	1	3	16	8.0
Total (W+X+U)	43	15	13	71	**35.5**

Table 5: Reasons for wrong (W), inexact (X) and unsupported (U) answers.

The system's major flaw lays in extracting the candidate answers: when it fails, the extraction patterns are either too lenient, causing overextraction (e.g. *origem da FAO* instead of *FAO* was the selected answer to question 84 "Como se chama a Organização para a Alimentação e Agricultura das Nações Unidas?" [What is the Food and Agriculture Organization of the United Nations called?]) or too strict, causing underextraction (e.g., *porta-voz* instead of *porta-voz do papa João Paulo 2º* was the answer to question 166 "Quem é Joaquín Navarro-Valls?" [Who is Joaquín Navarro-Valls?]). Additionally, the system is also not ready to cope with questions that should return a list or coordinated terms, like question 22 "Que dois cientistas descobriram as proteínas G?" [Which two scientists discovered G proteins?], that seeks coordinate terms, but the extraction allows only retrieving a single answer. Anaphora resolution and setting the value of k to 1, instead of 0 (see subsection 3.4) to check the answer in close sentences, could improve the system's performance when processing questions like question 17 "Quantas pessoas vivem nas ilhas Aaland?" [How many people live in the Aaland islands?].

The second major flaw is the way the system handled NIL questions: from the 18 questions that should have returned a NIL string, the system only returned 2 of them correctly. NIL recall is quite low (11%) because we do not actually measure a confidence score to decide if an answer is good enough. Frequently, the answer sentence matches only one pivot of a question, which sometimes is too weak a match. On the other hand, we do not demand exclusivity for some question categories. For example, a question beginning with "Qual é a altura de..." [What is the

height of...] should not have another category of question besides <DIMENSION>, which demands a numeric answer with an appropriate measure unit. Nevertheless, performing NIL validation may lead to discard correct but somehow weakly supported answers; a compromise of strictness is needed in the implementation of such an algorithm.

The third flaw has to do with the choice of the final answer, i.e., with the algorithm that calculates the final scores of the candidate answers (see subsection 3.5). Occasionally, the correct answer is ranked in the second position right after the wrong answer that was chosen (e.g. *companhia aérea belga*, the correct answer to question 21 "O que é a Sabena?" [What is Sabena?], followed the selected answer *Swissair*). Not very frequently, the system had to choose between answers equally scored (e.g. *presidente* and *presidente filipino* had the same exact score, but it was the first one (inexact) that was selected as the answer to question 165 "Quem é Fidel Ramos?" [Who is Fidel Ramos?]).

The last flaw reveals that the system sometimes misses the document containing the answer, during the document retrieval stage. Because that document will never be analysed, this failure is unrecoverable. This is a rare source of error, though, as the statistics of 5 show. One instance of this problem happened in question 85 "Diga o nome de um assassino em série americano." [Name an American serial killer]. During document retrieval, the system was not able to establish a relation between *americano* [American] and *EUA* [USA]. Therefore, it did not retrieve the document containing the sentence with the correct answer (*John Wayne Gacy*): "Estava marcada para hoje em Chicago à 0h01 local (2h01 em Brasília) a execução de John Wayne Gacy, maior assassino em série da história dos EUA." Another instance of this problem occurred with question 30 "Que percentagem de crianças não tem comida suficiente no Iraque?" [What percentage of children does not have enough food in Irak?]. Here, the system did not retrieve the sentence containing the answer (*entre 22 e 30 por cento*): "Os salários não têm acompanhado a subida da inflação e as agências humanitárias advertiram que entre 22 e 30 por cento das crianças iraquianas estão gravemente mal nutridas." In this case, the query expansion allowed by the indexation of the heads of derivation enabled the use of the gentilic information of the entries (inhabitant/country) to relate *iraquianas* [Iraqis] to *Iraque* [Iraq] but was not able to establish a synonymic relation between *não tem comida suficiente* [does not have enough food] and *mal nutridas* [badly nourished]. One way to obviate this is to increase the factor K_O in (2), when comparing the ontology domains of the question with those of the documents. In this particular case, we can see that the question and the answer sentence share a common domain: the words *comida* (question) and *nutridas* (answer) are grouped under the same level: metabolism/nutrition. This ontological information seems to be very helpful; however, since we use a low value for K_O we do not actually take full profit of it yet.

Consider now the run scores according to what kind of information questions ask for, as shown in Table 6.

Answer type	Right	Wrong	Inexact	Unsup.	**Total**	**Accuracy (%)**
LOCATION	28	6	0	1	35	80.0
MEASURE	11	7	0	0	18	61.1
ORGANIZATION	19	14	5	0	38	50.0
OTHER	12	8	1	0	21	57.1
PERSON	45	19	7	2	73	61.6
TIME	14	1	0	0	15	93.3

Table 6: Results by CLEF types of answer.

Crossing these CLEF answer types with our question categories, we found out that the best results were achieved by categories <DATE OF EVENT>, <DATE OF BIRTH> and <DATE OF DEATH> (type TIME of Table 6) and <LOCATION>, <TOWN>, <COUNTRY> (type LOCATION of Table 6). Interestingly, answer type PERSON congregates two of our question categories, namely <FUNCTION> and <DENOMINATION>. These two separated categories allow a more fine-grained search, since the <FUNCTION> category retrieves answers with names of professions (ergonyms)

or NEs like *President of the United States of America*, while the <DENOMINATION> category retrieves answers mainly with proper nouns.

Finally, we refer a special note on question 83 "Quem é Iqbal Masih." [Who is Iqbal Masih?]. This is a tricky question: it looks for a definition and the system retrieved *o rapazinho da foto* [the little boy in the photo]. Could that be considered a definition? What the user considers a definition may vary according to his/her information background. This answer was extracted from a standard apposition structure, however, in terms of meaning conveyed to the user, it may not be considered responsive enough. If the answer was extracted by a system that allows the user to visualize the document that contained the answer, as is the case of the TRUST search engine, then the answer *o rapazinho da foto* should be satisfactory. However, being the answer extracted by a system that does not allow the visualization of the document, it is not of great utility to the user. In this case, the system should have returned other (not easily extractable) answers in the same sentence, such as *quase-escravo* [almost slave], the more descriptive *peregrino pelo mundo em defesa de seis milhões de crianças que no Paquistão são exploradas por negociantes sem escrúpulos* [pilgrim over the world in defence of six million children who in Pakistan are exploited by unscrupulous business men] or even a summary of the whole document.

This evaluation furthermore showed that Brazilian Portuguese was not a relevant problem for a system that only used a European Portuguese lexicon. There were few questions with exclusive Brazilian spelling or Brazilian terms – 102 "Que vulcão teve uma erupção em *junho* de 1991?", 114 "Onde surgiu a *Aids*?", 124 "Em quantos filmes da série 'Superman' *estrelou* Chistopher Reeve?", 127 "Quantas repúblicas compunham a *Iugoslávia*?", 148 "Que *time* se mudou para Salt Lake City?", 183 "Quem é o *prefeito* de Lisboa?". The system was able to retrieve several correct answers from Brazilian target documents, as in the case of the answer *135 quilômetros* to question 132 "Que distância separa Cuba da Flórida?" [What distance separates Cuba from Florida?]). That was not the case, however, with question 151 "Em que posição joga Taffarel?" [In which position does Taffarel play?], whose expected answer *goleiro* [goalkeeper] was not recognised by the European Portuguese lexicon.

5 Conclusions and future work

Throughout this paper we accounted for the description and evaluation of Priberam's QA system. The results obtained in the QA@CLEF monolingual task by both Priberam (for Portuguese) and Synapse (for French), who based their systems on the NLP technology developed for TRUST search engine, seem to state that the choices made are in the right track.

The architecture of our system is similar to many others, yet it distinguishes itself by the indexation of morphologically disambiguated words at sentence level and by the query expansion using heads of derivation. The use of the workbench described in section 2, as well as its associated descriptive languages, allows an easy maintenance and coding of several NLP features, and this is probably a big advantage since it makes the system scalable.

Despite the encouraging results detailed in the previous section, the system still has a long way to go before it can be efficient in a generic environment. We have spotted some improvements to be implemented in a near future, namely concerning the question/answer matching mechanism, syntactic treatment of questions and answers, anaphora resolution and semantic disambiguation. We intend to exploit further the ontology's potential. It can be a very useful resource during the stages of document and sentence retrieval, since it may improve the weighting of the documents and sentences by introducing semantic knowledge. This implies performing document clustering based on the ontology domains as well as inferring from question analysis those that should be predominant in the target documents. Future work will also address the treatment of questions that should return a list and the refinement of the question answering system for Web searching.

Currently we are participating in M-CAST – Multilingual Content Aggregation System based on TRUST Search Engine – (EDC 22249 M-CAST), an European eContent project whose aim is the development of a multilingual platform to access and search large multilingual text collections, such as internet libraries, publishing houses resources, press agencies and scientific databases, etc.

This participation will lead to greater enhancements, especially on the extraction of answers from books, which may prove to be quite different from extracting from newspaper articles.

Acknowledgements

Priberam Informática would like to thank the partners of the NLUC consortium, namely Synapse Développement, for sharing its experience and knowledge, thus allowing us to compare and test our two similar but different approaches. Priberam would also like to express its thanks to the CLEF organization and to Linguateca for preparing and supervising the Portuguese evaluation. Finally, we would like to acknowledge the support of the European Commission in TRUST (IST-1999-56416) and M-CAST (EDC 22249 M-CAST) projects.

References

[1] D. Santos and P. Rocha. CHAVE: topics and questions on the Portuguese participation in CLEF. In C. Peters and F. Borri, editors, *Cross Language Evaluation Forum: Working Notes for the CLEF 2004 Workshop (Bath, UK, 15-17 September)*, pages 639–648, 2004. Also available at `http://www.clef-campaign.org/2004/working_notes/WorkingNotes2004/76.PDF.`

[2] P. Quaresma, L. Quintano, I. Rodrigues, J. Saias, and P. Salgueiro. The University of Évora approach to QA@CLEF-2004. In C. Peters and F. Borri, editors, *Cross Language Evaluation Forum: Working Notes for the CLEF 2004 Workshop (Bath, UK, 15-17 September)*, pages 403–411, 2004.

[3] L. Costa. First evaluation of Esfinge – a question-answering system for Portuguese. In C. Peters and F. Borri, editors, *Cross Language Evaluation Forum: Working Notes for the CLEF 2004 Workshop (Bath, UK, 15-17 September)*, pages 393–402, 2004.

[4] C. Amaral, D. Laurent, A. Martins, A. Mendes, and C. Pinto. Design and Implementation of a Semantic Search Engine for Portuguese. In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal, 26-28 May*, volume 1, pages 247–250, 2004. Also available at `http://www.priberam.pt/docs/LREC2004.pdf.`

[5] D. Laurent, M. Varone, C. Amaral, and P. Fuglewicz. Multilingual Semantic and Cognitive Search Engine for Text Retrieval Using Semantic Technologies. In *Pre-proceedings of the 1st Workshop on International Proofing Tools and Language Technologies (Patras, Greece, 1-2 July)*, 2004.

[6] C. Amaral, H. Figueira, A. Mendes, P. Mendes, and C. Pinto. A Workbench for Developing Natural Language Processing Tools. In *Pre-proceedings of the 1st Workshop on International Proofing Tools and Language Technologies (Patras, Greece, 1-2 July)*, 2004. Also available at `http://www.priberam.pt/docs/WorkbenchNLP.pdf`.

[7] D. Laurent, P. Séguéla, and S. Nègre. Cross Lingual Question Answering using QRISTAL for CLEF 2005. In *Working Notes for the CLEF 2005 Workshop, 21-23 September, Wien, Austria*, 2005. To appear.

[8] S.M. Thede and M.P. Harper. A second-order hidden Markov model for part-of-speech tagging. In *Proceedings of the 37th Annual Meeting of the ACL, Maryland: College Park*, pages 175–182, 1999.

[9] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing (2nd printing)*. The MIT Press, Cambridge, Massachusetts, 2000.

[10] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag New York, Inc., 2001.

[11] C. Monz and M. de Rijke. The University of Amsterdam's textual question answering system. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Tenth Text Retrieval Conference (TREC 2001), Gaithersburg, Maryland, 13-16 November*, pages 519–528, 2002.

[12] D. Ferrés, S. Kanaan, E. González, A. Ageno, H. Rodríguez, M. Surdeanu, and J. Turmo. TALP-QA System at TREC 2004: Structural and Hierarchical Relaxing of Semantic Constraints. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004), Gaithersburg, Maryland, 16-19 November*, 2005. To appear.

[13] B. Magnini, A. Vallin, C. Ayache, G. Erbach, A. Peñas, M. de Rijke, P. Rocha, K. Simov, and R. Sutcliffe. Overview of the CLEF 2004 multilingual question answering track. In C. Peters and F. Borri, editors, *Cross Language Evaluation Forum: Working Notes for the CLEF 2004 Workshop (Bath, UK, 15-17 September)*, pages 281–294, 2004.

[14] K. Lavenus, J. Grivolla, L. Gillard, and P. Bellot. Question-answer matching: two complementary methods. In *Proceedings of RIAO 2004, University of Avignon (Vaucluse), France*, 2004.

[15] Xin Li and D. Roth. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan*, 2002.

[16] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.

[17] E.M. Voorhees. Overview of the TREC 2004 Question Answering Track. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004), Gaithersburg, Maryland, 16-19 November*, 2005. To appear. Also available at `http://trec.nist.gov/pubs/trec13/papers/QA.OVERVIEW.pdf`.